KROME: a first view into the package

Download, explore, and prepare a network file

September 17, 2014

Stefano Bovino

KROME school 2014

Historische Sternwarte-Göttingen

Institute for Astrophysics
Georg-August Universität Göttingen
Germany

KROME

better science through chemistry

Aims and goals

what we will learn during this talk (hopefully!)

- what the hell is KROME?
- How KROME is structured
- Git and direct download
- How to run KROME
- How to prepare a chemical network (tokens)
- Quick overview of the pre-built networks and tests
- Something about the KROME_USER module

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*
- B: *what kind of cooling?*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*
- B: *what kind of cooling?*
- A: *mmmh... something to form stars*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*
- B: *what kind of cooling?*
- A: *mmmh... something to form stars*
- B: *do you know how much intricated is the microphysics?*

Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*
- B: *what kind of cooling?*
- A: *mmmh... something to form stars*
- B: *do you know how much intricated is the microphysics?*
- A: *not really, but I'd like to form stars!*

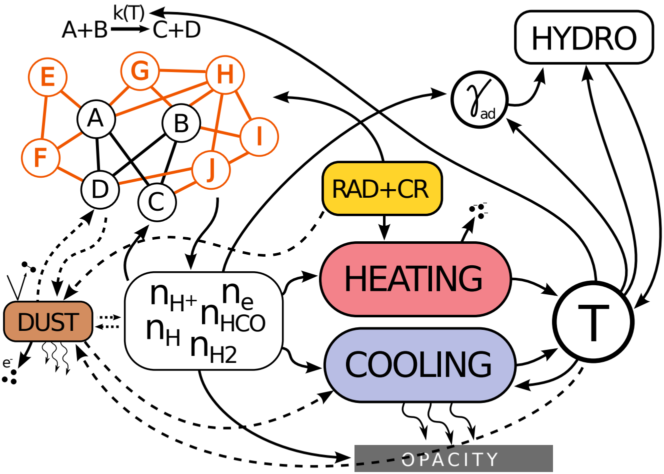
Why KROME?

Overheard dialougue between two astrophysicists:

- A: *I'd like to include chemistry in my simulation*
- B: *what do you exactly need?*
- A: *I'd like some cooling*
- B: *what kind of cooling?*
- A: *mmmh... something to form stars*
- B: *do you know how much intricated is the microphysics?*
- A: *not really, but I'd like to form stars!*
- ... and then B shows to A the following plot!

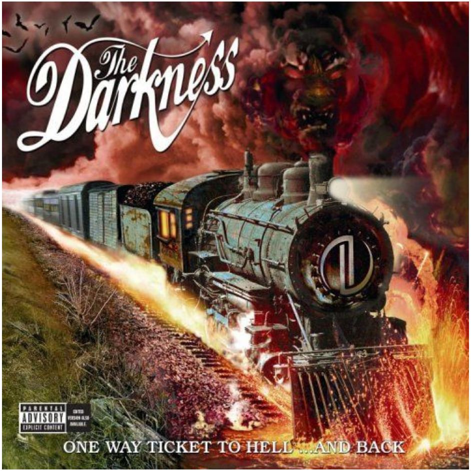
Why KROME?

welcome to the hell



Welcome to the hell!

KROME school 2014

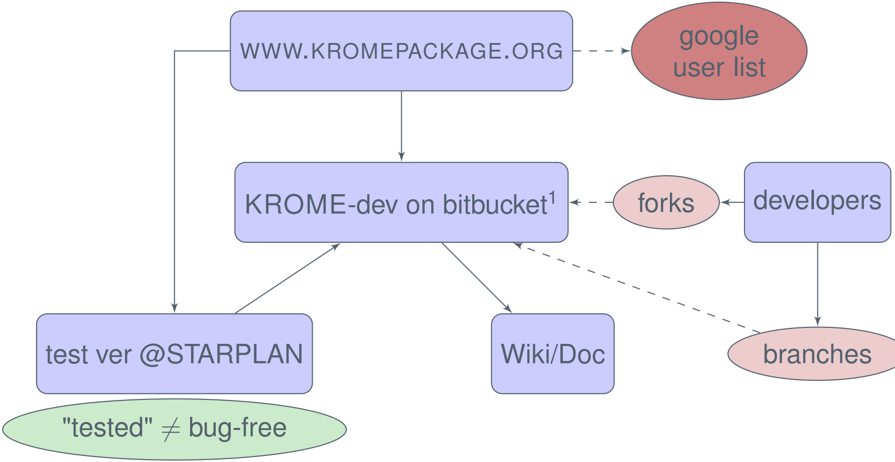

The Darkness
PARENTAL ADVISORY EXPLICIT CONTENT
EDITED VERSION ALSO AVAILABLE
ONE WAY TICKET TO HELL ...AND BACK

Things to know about KROME

- the core of KROME is a mix of PYTHON and FORTRAN 90.
- KROME is a pre-processor
- to use KROME in your simulations you need to run ./krome first
- it creates all the machinery (fortran 90 routines) needed to solve thermal/chemical evolution of the gas
- KROME reduces the hard-coding from the users

Python is the "most powerful language you can still read"

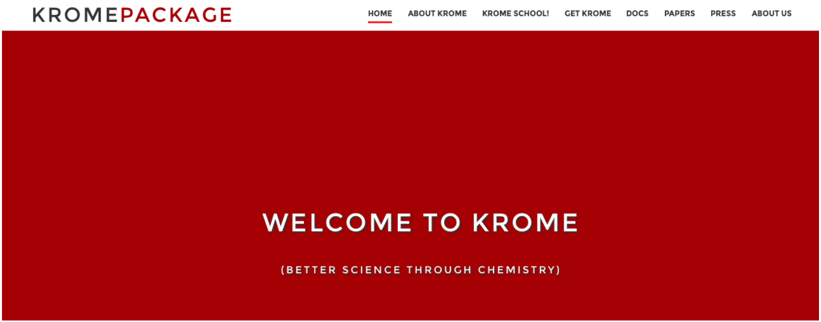
Organization and structure
KROME on the www



[1] https://bitbucket.org/tgrassi/krome

How to download KROME
Via the website

- simply go into www.kromepackage.org



KROMEPACKAGE

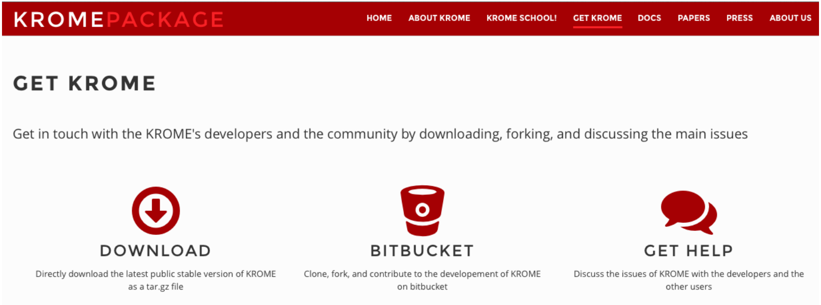
HOME ABOUT KROME KROME SCHOOL! GET KROME DOCS PAPERS PRESS ABOUT US

WELCOME TO KROME

(BETTER SCIENCE THROUGH CHEMISTRY)

How to download KROME
via the website

- simply go into www.kromepackage.org



GET KROME

Get in touch with the KROME's developers and the community by downloading, forking, and discussing the main issues

DOWNLOAD

Directly download the latest public stable version of KROME as a tar.gz file

BITBUCKET

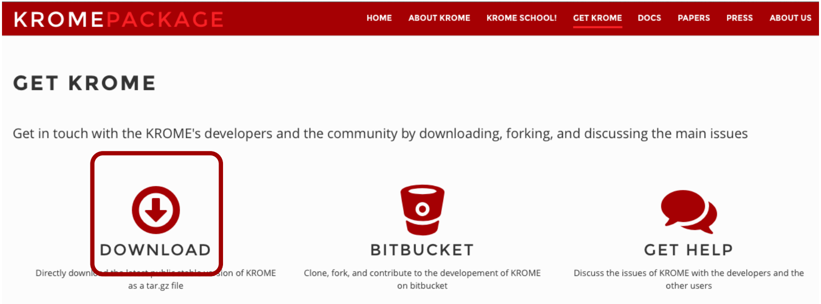
Clone, fork, and contribute to the developement of KROME on bitbucket

GET HELP

Discuss the issues of KROME with the developers and the other users

How to download KROME
via the website

- simply go into www.kromepackage.org
- you’ll open the krome test web page
- this contains info on the test status which determines the "tested" version



GET KROME

Get in touch with the KROME's developers and the community by downloading, forking, and discussing the main issues

DOWNLOAD

Directly download the latest public stable version of KROME as a tar.gz file

BITBUCKET

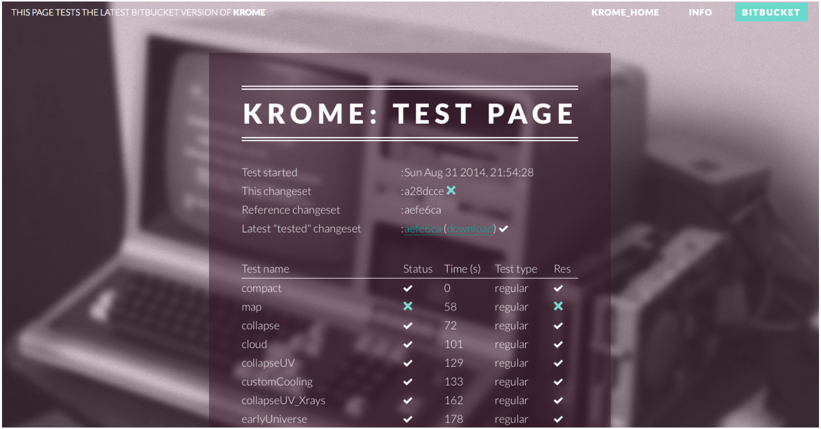
Clone, fork, and contribute to the developement of KROME on bitbucket

GET HELP

Discuss the issues of KROME with the developers and the other users

How to download KROME

- from the website you can access the repository
- a version of KROME is "tested" if the test called "regular" are working



KROME: TEST PAGE

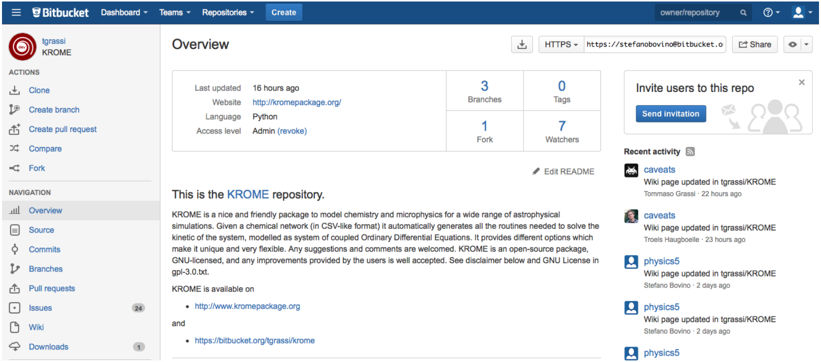
Test started :Sun Aug 31 2014, 21:54:28
This changeset :a28dcce ✗
Reference changeset :aefe6ca
Latest "tested" changeset :aefe6ca (download) ✓

Test name	Status	Time (s)	Test type	Res
compact	✓	0	regular	✓
map	✗	58	regular	✗
collapse	✓	72	regular	✓
cloud	✓	101	regular	✓
collapseUV	✓	129	regular	✓
customCooling	✓	133	regular	✓
collapseUV_Xrays	✓	162	regular	✓
earlyUniverse	✓	178	regular	✓

How to download KROME
via bitbucket

requirements:

- basic knowledge of the **git** commands
    - if NOT you will learn something during the school (last talk)
- in any case go into https://bitbucket.org/tgrassi/krome



Overview

Last updated 16 hours ago
Website http://kromepackage.org/
Language Python
Access level Admin (revoke)

3 Branches · 0 Tags · 1 Fork · 7 Watchers

Invite users to this repo

This is the KROME repository.

KROME is a nice and friendly package to model chemistry and microphysics for a wide range of astrophysical simulations. Given a chemical network (in CSV-like format) it automatically generates all the routines needed to solve the kinetic of the system, modelled as system of coupled Ordinary Differential Equations. It provides different options which make it unique and very flexible. Any suggestions and comments are welcomed. KROME is an open-source package, GNU-licensed, and any improvements provided by the users is well accepted. See disclaimer below and GNU License in gpl-3.0.txt.

KROME is available on

- http://www.kromepackage.org

and

- https://bitbucket.org/tgrassi/krome

caveats
Wiki page updated in tgrassi/KROME
Tommaso Grassi · 22 hours ago

caveats
Wiki page updated in tgrassi/KROME
Troels Haugboelle · 23 hours ago

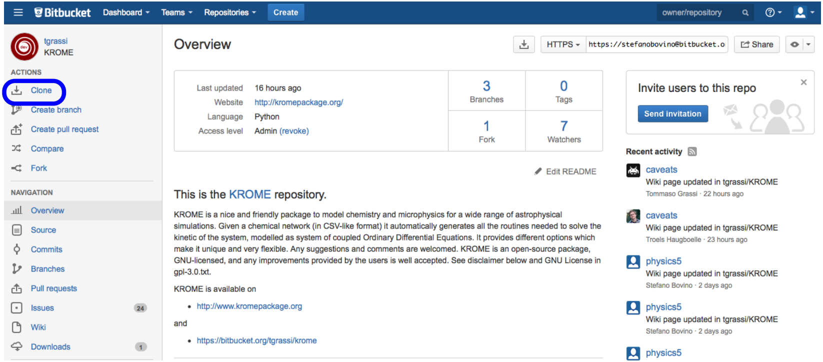
physics5
Wiki page updated in tgrassi/KROME
Stefano Bovino · 2 days ago

physics5
Wiki page updated in tgrassi/KROME
Stefano Bovino · 2 days ago

physics5

How to download KROME

via bitbucket



Overview

Last updated	16 hours ago
Website	http://kromepackage.org/
Language	Python
Access level	Admin (revoke)

3 Branches	0 Tags
1 Fork	7 Watchers

This is the KROME repository.

KROME is a nice and friendly package to model chemistry and microphysics for a wide range of astrophysical simulations. Given a chemical network (in CSV-like format) it automatically generates all the routines needed to solve the kinetic of the system, modelled as system of coupled Ordinary Differential Equations. It provides different options which make it unique and very flexible. Any suggestions and comments are welcomed. KROME is an open-source package, GNU-licensed, and any improvements provided by the users is well accepted. See disclaimer below and GNU License in gpl-3.0.txt.

KROME is available on

- http://www.kromepackage.org

and

- https://bitbucket.org/tgrassi/krome

caveats
Wiki page updated in tgrassi/KROME
Tommaso Grassi · 22 hours ago

caveats
Wiki page updated in tgrassi/KROME
Troels Haugboelle · 23 hours ago

physics5
Wiki page updated in tgrassi/KROME
Stefano Bovino · 2 days ago

physics5
Wiki page updated in tgrassi/KROME
Stefano Bovino · 2 days ago

physics5

How to download KROME

via bitbucket

```
$ git clone https://bitbucket.org/tgrassi/krome.git
Cloning into 'krome'...
remote: Counting objects: 4193, done.
remote: Compressing objects: 100% (2063/2063),
    done.
remote: Total 4193 (delta 2675), reused 3278
    (delta 2094)
Receiving objects: 100% (4193/4193), 16.81 MiB |
    673.00 KiB/s, done.
Resolving deltas: 100% (2675/2675), done.
Checking connectivity... done.
```

- you will generate a folder named "krome"
- note that you're downloading the "dev" version

Explore KROME

through the files

```
$ ls
README.md       data            options_example   tests
alltest.py      gpl-3.0.txt     outtest.md5       tools
argparse.py     krome           patches           wizard
build           kromelib.py     solver
changelog.txt   kromeobj.py     src
clean           networks        test_list
```

- folders
- python files-the core of the package
- executable-pre-processor
- other files (e.g. clean to clean up your working dir)

Explore KROME
through the folders

Folders names are self-explanatory

- BUILD → contains the fortran routines
- DATA → cross-sections, cooling tables...
- NETWORKS → all the pre-built networks
- PATCHES → 3D hydro-code patches source routines
- SOLVER → chemical solvers routines
- SRC → fortran sources routines/need to be pre-processed
- TESTS → contains the pre-built tests (ready to run)
- TOOLS → a series of useful tools (e.g. databases converter, for advanced users)

Some simple python files: just to know

- ALLTEST.PY → run all the pre-built tests and check if they work
- ARGPARSE.PY → needed if you don't have it installed

Explore KROME
the first run

several ways to run KROME:

- from shell using in-line commands
- via the helper-script "wizard"
- by using an option file

KROME has internal check to verify the options

```
$ ./krome
******************************
     WELCOME TO KROME
******************************

ERROR: you must define -n FILENAME or -network
    FILENAME, where FILENAME is the reaction file!
```

The first run
using in-line commands

It requires you to know the options!

KROME help in unix format

```
$ ./krome -h
******************************
    WELCOME TO KROME
******************************

KROME a package for astrochemistry and microphysics

optional arguments:
  -h, --help show this help message and exit
  -ATOL ATOL set solver absolute tolerance to the float
      or double value ATOL
  -C create a simple C wrapper
  -n FILENAME reaction network file
  -network FILENAME same as -n
  -nochargeCheck skip reaction charge check
```

The first run

using wizard

If you don’t know how to start you can use the "wizard" helper

```
$ ./wizard
********************
 option wizard!
********************
- Path of your chemical network [networks/react_COthin]:
```

The first run
using wizard

It will ask questions about the possible options you can enable

```
$ ./wizard
********************
 option wizard!
********************
- Path of your chemical network
    [networks/react_COthin]: networks/react_primordial
added -n=networks/react_primordial

- Use number density (otherwise mass fractions)? [y]:
```

The first run

using wizard

```
0) NONE (No cooling)
1) ATOMIC (Atomic from Cen 1992)
2) H2 (H2 from Glover+2007)
3) HD (HD from Lipovka+2007)
4) DH (Endothermic with thermochemical data)
5) DUST (Dust cooling)
6) H2GP98 (H2 from Galli+Palla 1998)
7) COMPTON (Compton)
8) EXPANSION (Isothermal expanding gas)
9) CIE (Collisional induced)
10) CONT (Continuum emission)
11) CHEM (Endothermic reactions)
...
21) FeII (FeII cooling)
- Cooling functions (use numbers above comma
    separated)? [0]:
```

The first run
using wizard

- at the end of the process you will get the in-line commands to correctly run KROME
- additionally you will have an "options.kop" file in your folder

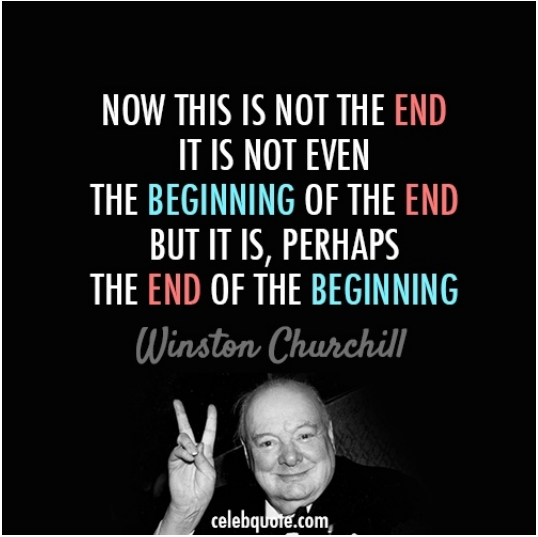
```
Your call to krome is:
 ./krome -n=networks/react_primordial -useN
or use the option file
 options.kop

_________________________________
 ./krome -options = options.kop
```

Note: during the school exercise we will use the interactive approach

Summarizing

- use in-line commands (need to know the options, check -h, -help)
- use the wizard helper (much easier)
- use the option file provided into the KROME folder (follow the option_example)

Whatever you decide to use KROME will run!



NOW THIS IS NOT THE END
IT IS NOT EVEN
THE BEGINNING OF THE END
BUT IT IS, PERHAPS
THE END OF THE BEGINNING

Winston Churchill

celebquote.com

Chemical Network files

quick overview

YOU NEED A CHEMICAL NETWORK

Three different approaches

- use one of the pre-built KROME networks (under networks/ folder)
- use the KIDA → KROME converter tool to select reactions from standard database
- prepare your own networks based on databases, literature, and/or pre-existing networks

Chemical Network files

quick overview

YOU NEED A CHEMICAL NETWORK

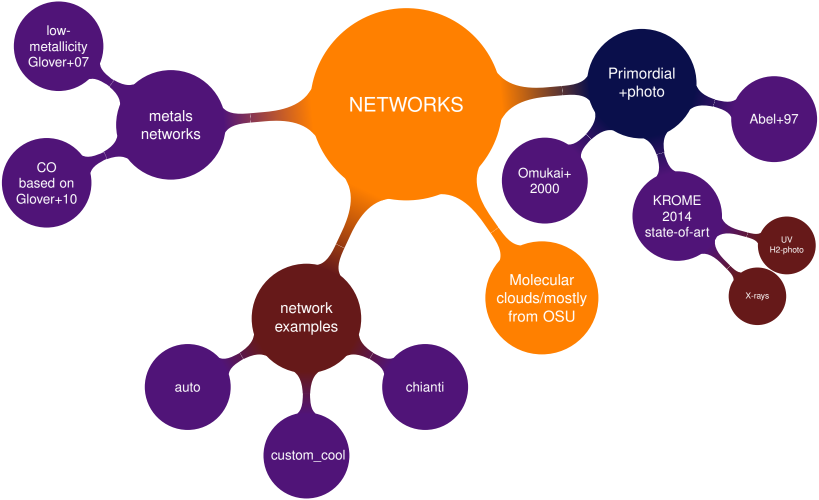
Three different approaches

- use one of the pre-built KROME networks (under networks/ folder)
- use the KIDA → KROME converter tool to select reactions from standard database
- prepare your own networks based on databases, literature, and/or pre-existing networks

Pre-built networks

a graphical overview

+other networks



Chemical Network files

quick overview

YOU NEED A CHEMICAL NETWORK

Three different approaches

- use one of the pre-built KROME networks (under networks/ folder)
- use the KIDA → KROME converter tool to select reactions from standard database → look at the Wiki
- prepare your own networks based on databases, literature, and/or pre-existing networks

Chemical Network files

quick overview

YOU NEED A CHEMICAL NETWORK

Three different approaches

- use one of the pre-built KROME networks (under networks/ folder)
- use the KIDA → KROME converter tool to select reactions from standard database
- prepare your own networks based on databases, literature, and/or pre-existing networks → use of the KROME-tokens

Chemical network
without tokens

- KROME networks files are written in a simple Comma-Separated Values (CSV) format
- you can write simple networks without "tokens" by using the default format:

$$R_1 + R_2 + R_3 \xrightarrow{k[T_{min}, T_{max}]} P_1 + P_2 + P_3 + P_4 \qquad (1)$$

Let's assume we want to implement the following network:

- $H_2^+ + H^- \rightarrow H + H_2$
- $H_2^+ + H \rightarrow H^+ + H_2$
- $H^- + H^+ \rightarrow e^- + H_2^+$

Chemical network

without tokens

- $H_2^+ + H^- \rightarrow H + H_2 \qquad k_1 = 5 \times 10^{-7}\sqrt{100/T_{gas}}$
- $H_2^+ + H \rightarrow H^+ + H_2 \qquad k_2 = 6 \times 10^{-10}$
- $H^- + H^+ \rightarrow e^- + H_2^+ \qquad k_3 = 10^{-8}T_{gas}^{-0.4}$

first_network

```
#Dalgarno & Lepp 1987
1,H2+,H-,,H,H2,,,NONE,NONE,5.d-7*sqrt(1.d2*invT)

//Karpas 1979
2,H2+,H,,H+,H2,,,NONE,NONE,6.0d-10

/*Poulart 1978*/
3,H-,H+,,E,H2+,,,NONE,NONE,1.d-8*Tgas**(-0.4d0)
```

Run KROME

analysing the out

```
$ ./krome -n networks/first_network
******************************
    WELCOME TO KROME
******************************

************************************************
WARNING: the folder build/ is not empty
 some items may be replaced. Do you want to
    proceed?
To avoid this message use -unsafe option.
************************************************
```

Run KROME

analysing the out

Network composed by: H_2^+, H_2, H^-, e^-, H^+, H

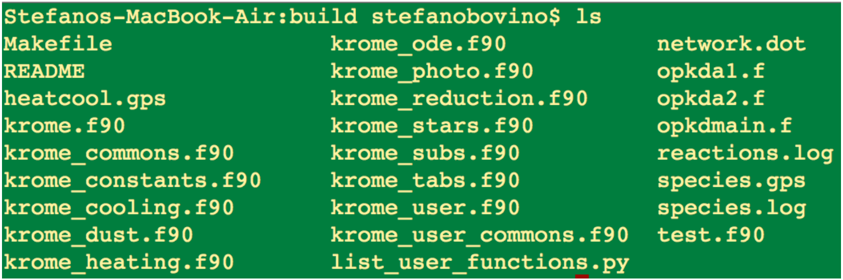
```
ODEs needed: 10
Reactions found: 3
Species found: 6
Species list saved in build/species.log
Species index initialization for gnuplot in
   build/species.gps
Heating cooling index init for gnuplot in
   build/heatcool.gps
Reactions saved in build/reactions.log
ODE partition: [6 atom/mols] + [1 CR] + [1 PHOT] +
   [1 Tgas] + [1 dummy] = 10 ODEs
ODEs list: H2+, H, H2, H+, E, H-, CR, g, Tgas,
   dummy
```

Run KROME
analysing the out

```
Jacobian non-zero elements: 19 over 100
(19.0% of total elements, sparsity = 81.0%)
solver info:
 MF: 222
 MOSS+METH+MITER: 2+2+2
 LWM: 312 LRW: 422
 ...
 Prepearing files in /build...
- writing krome_commons.f90... done!
...
******************
Everything done, goodbye!
*******************************************
41. COMPUTERS ARE LIKE OLD TESTAMENT GODS;
LOTS OF RULES AND NO MERCY
--- Joseph Campbell
*******************************************
```

Access the build folder

files generated by KROME



```
Stefanos-MacBook-Air:build stefanobovino$ ls
Makefile                krome_ode.f90           network.dot
README                  krome_photo.f90         opkda1.f
heatcool.gps            krome_reduction.f90     opkda2.f
krome.f90               krome_stars.f90         opkdmain.f
krome_commons.f90       krome_subs.f90          reactions.log
krome_constants.f90     krome_tabs.f90          species.gps
krome_cooling.f90       krome_user.f90          species.log
krome_dust.f90          krome_user_commons.f90  test.f90
krome_heating.f90       list_user_functions.py
```

Look into the modules

krome_ode module

```
k(:) = coe_tab(n(:)) !compute coefficients

   !H2+
   dn(1) = &
      -k(1)*n(idx_H2j)*n(idx_Hk) &
      -k(2)*n(idx_H2j)*n(idx_H) &
      +k(3)*n(idx_Hk)*n(idx_Hj)

   !H-
   dn(2) = &
      -k(1)*n(idx_H2j)*n(idx_Hk) &
      -k(3)*n(idx_Hk)*n(idx_Hj)

   !H2
   dn(3) = &
      +k(1)*n(idx_H2j)*n(idx_Hk) &
      +k(2)*n(idx_H2j)*n(idx_H)
```

Look into the modules

krome_main module

```
!****************************
   !init DLSODES (see DLSODES manual)
   neq = nspec !number of eqns
   liw = size(iwork)
   lrw = size(rwork)
   iwork(:) = 0
   rwork(:) = 0.d0
   itol = 4 !both tolerances are arrays
   rtol(:) = 1.000000d-04 !relative tolerance
   atol(:) = 1.000000d-20 !absolute tolerance
   icount_max = 100 !maximum number of iterations
  itask = 1
   iopt = 0

   !MF=
   ! = 222 internal-generated JAC and sparsity
   ! = 121 user-provided JAC and internal generated sparsity
   ! = 22 internal-generated JAC but sparsity user-provided
   ! = 21 user-provided JAC and sparsity
   MF = 222
   !end init DLSODES
```

Use KROME

krome call

```
Call KROME in your code as:
    call krome(x(:), gas_density, gas_temperature,
        time_step)
where:
 x(:) is a real*8 array of size 6 of the mass
     fractions
 gas_density is the gas density in [g/cm3]
 gas_temperature is the gas temperature in [K]
 time_step is the integration time-step in [s]
```

- We're using mass fraction.
- if you want to use number densities enable "-useN" option

Additional useful files

In addition KROME creates the following files:

- species.log and reactions.log
  - the first one contains the list of species with their indexes (e.g. useful for initialization)
  - the second ones contains the list of reactions included in the network
- gnuplot scripts (to be used for testing, not in hydro-code)
  - heatcool.gps (to plot the cooling/heating contributions)
  - species.gps (to plot the evolution of the species)
- network.dot (next slide)

The species.log file

```
#This file contains a list of the species used with their
    indexes

1 H idx_H
2 E idx_E
3 H+ idx_Hj
4 HE idx_HE
5 HE+ idx_HEj
6 HE++ idx_HEjj
7 H- idx_Hk
8 H2 idx_H2
9 H2+ idx_H2j
10 CR idx_CR
11 g idx_g
12 Tgas idx_Tgas
13 dummy idx_dummy
```

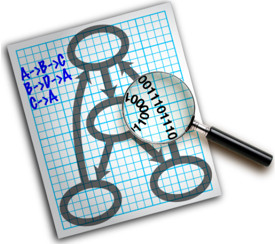
if you want to initialize H in your code

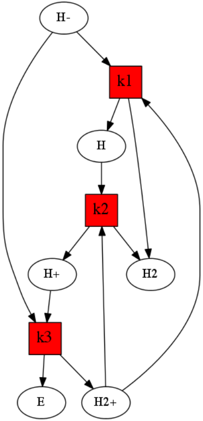
x(krome_idx_H) = VALUE

Nice tool
graphviz

Once KROME runs it generates a file "network.dot" which can be easily plotted with Graphviz.

```
dot -Tpng network.dot > output.png
```





- $H_2^+ + H^- \rightarrow H + H_2$
- $H_2^+ + H \rightarrow H^+ + H_2$
- $H^- + H^+ \rightarrow e^- + H_2^+$

Short summary

what we know and what we don't

What we know at this stage:

- how KROME is structured (folders/files etc.)
- how to run KROME (in-line commands, wizard, options file)
- how to prepare a simple network by using the default format
- how it looks the KROME output (at least partially)
- what is inside build (utilities/tools/routines)

What we don't know yet

- how to use the KROME-tokens to prepare a network
- how it is organized the krome_user module
- how to call KROME from a "framework" code
- how to run a built-in test

Tokens

prepare an advanced chemical network

KROME includes a series of useful tokens which simplify and help the preparation of the chemical network

In general

- A token is a string of one or more characters that is significant as a group
- The process of forming tokens from an input stream of characters is called tokenization
- Tokens are identified based on the specific rules of the lexer (or parser)

In KROME

the tokens are identified by the delimiter "@" (this procedure should not be confused with python tokenize)

Tokens

cont'd

- @format
- @var
- @common
- @noTabNext, @noTab_start and @noTab_end
- @CR_start and @CR_stop
- @Xray_start and @Xray_stop
- @photo_start and @photo_stop
- @ghost
- @tabvar
- @reactionModifier_start and @reactionModifier_stop
- @ODEModifier_start and @ODEModifier_stop
- @cooling_start and @cooling_stop

Tokens

cont'd

- @format
- @var
- @common
- @noTabNext, @noTab_start and @noTab_end
- @CR_start and @CR_stop
- @Xray_start and @Xray_stop
- @photo_start and @photo_stop
- @ghost
- @tabvar
- @reactionModifier_start and @reactionModifier_stop
- @ODEModifier_start and @ODEModifier_stop
- @cooling_start and @cooling_stop

Tokens
@format

@format to define a different format for the rates definition (e.g. two reactants-one product, no temperature limits, and so on).

- $H_2^+ + H^- \rightarrow H + H_2 \qquad k_1 = 5\times10^{-7}\sqrt{100/T_{gas}}$
- $H_2^+ + H \rightarrow H^+ + H_2 \qquad k_2 = 6\times10^{-10}$
- $H^- + H^+ \rightarrow e^- + H_2^+ \qquad k_3 = 10^{-8}T_{gas}^{-0.4}$

first_network, default format:
IDX,R,R,R,P,P,P,P,TMIN,TMAX,RATE

```
#Dalgarno & Lepp 1987
1,H2+,H-,,H,H2,,,NONE,NONE,5.d-7*sqrt(1.d2*invT)

#Karpas 1979
2,H2+,H,,H2,H+,,,NONE,NONE,6.0d-10

#Poulart 1978
3,H-,H+,,H2+,E,,,NONE,NONE,1.d-8*Tgas**(-0.4d0)
```

Tokens
@format

first_network

```
#Dalgarno & Lepp 1987
@format:idx,R,R,P,P,rate
1,H2+,H-,H,H2,5.d-7*sqrt(1.d2*invT)

#Karpas 1979
2,H2+,H,H2,H+,6.0d-10

#Poulart 1978
3,H-,H+,H2+,E,1.d-8*Tgas**(-0.4d0)
```

Now let's add a three-body reaction:

- $H_2 + H_2 \rightarrow H + H + H_2 \qquad k_4 = kh_{21}^{(1-a_{21})} kl_{21}^{a_{21}}$
- a_{21}, kh_{21}, kl_{21} very weird formulae!

First rule of chemical networks

Functional forms of the rates are always ugly

Tokens (cont'd)

@var and @noTabNext

```
#Dalgarno & Lepp 1987
@format:idx,R,R,P,P,rate
1,H2+,H-,H,H2,5.d-7*sqrt(1.d2*invT)

#Karpas 1979
2,H2+,H,H2,H+,6.0d-10

#Poulart 1978
3,H-,H+,H2+,E,1.d-8*Tgas**(-0.4d0)

#Omukai 2001
@format:idx,R,R,P,P,P,rate
@var:Hnuclei = get_Hnuclei(n(:))
@var:kl21 = 1.18d-10*exp(-6.95d4*invT)
@var:kh21 =
    8.125d-8*T**(-0.5d0)*exp(-5.2d4*invT)*(1.d0-exp(-6d3*invT))
@var:ncr21 =
    1d1**(4.845d0-1.3d0*log10(T*1d-4)+1.62d0*log10(T*1d-4)**2)
@var:a21=1.d0/(1.d0+(Hnuclei/ncr21))
@noTabNext
4,H2,H2,H,H,H2,kh21**(1.-a21)*kl21**a21
```

Tokens (cont'd)

@var and @noTabNext

- @var needed to define new variables
- @noTabNext tells KROME that this rate shouldn't be tabulated
  - valid if you enable the tabulation of the rate coefficients (-useTabs)
  - to be applied if your rates depend on other variables (e.g. density)

Now let's introduce a rate which depends on a common variable and on some particular function, e.g.

$$\mathrm{H_2} + \gamma \rightarrow \mathrm{H} + \mathrm{H} \tag{2}$$

Let's assume that the photodissociation rate depends on the flux intensity J_{21} and a self-shielding function.

$$k_{ph} = 9 \times 10^{-13} J_{21} f_{shield}(n(H_2), T_{gas}) \tag{3}$$

where $f_{shield} = 1$ means no shielding (stronger photodissociation)

Tokens (cont’d)

@commons

```
#Dalgarno & Lepp 1987
@format:idx,R,R,P,P,rate
1,H2+,H-,H,H2,5.d-7*sqrt(1.d2*invT)

#Karpas 1979
2,H2+,H,H2,H+,6.0d-10

#Poulart 1978
3,H-,H+,H2+,E,1.d-8*Tgas**(-0.4d0)

#H2 photodissociation H2 based on a BB spectrum at T=1d5 K
@format:idx,R,P,P,rate
@common:  user_J21
4. H2,H,H,9.d-13*user_J21*user_fshield(n,Tgas)
```

- all the commons from user should start with "user" (e.g. user_J21) to avoid internal conflicts
- the function "user_fshield(n,Tgas)" should be defined by the users in the module "krome_user_commons" (see the afternoon exercise).

The user module

krome_user

Once the user define a common variable (or in other cases) is possible to assign a value to that variable from the framework code by using the user functions provided by KROME.

standard structure

- CALL KROME_SET_USER_VARIABLE(VALUE)
- MY_VARIABLE = KROME_GET_USER_VARIABLE()

For example:

- CALL KROME_SET_USER_J21(0.1) → set the common user_J21
- J21 = KROME_GET_USER_J21() → assign the value to your variable named J21

The user module

krome_user

A list of the functions available in the krome_user module for the given network can be obtained by running the python script provided in build/

```
$ python list_user_functions.py

1) function krome_conserve(x,xi)
 alias for conserve in krome_subs

2) function krome_get_Tcmb()
  [no comments available]

3) function krome_get_user_J21()
  [no comments available]

4) subroutine krome_set_Tcmb(arg)
  [no comments available]

5) subroutine krome_set_user_J21(argset)
  [no comments available]
```

Built-in tests

overview

Under the folder "tests" you can find the following directories

- auto
- chianti
- cloud
- collapse
- collapseCO
- collapseUV
- collapseUV_Xrays
- collapseZ
- collapseZ_UV
- collapseZ_induced
- compact
- customCooling
- dust
- earlyUniverse
- hello
- lamda
- lotkav
- map
- reverse
- shock1D
- shock1Dcool
- shock1Dphoto
- slowmanifold
- stars
- wrapC

Built-in tests
overview

something to know

- mostly 0D and 1D
- some of the tests are stand-alone codes that can be used for applications, they are based on existing benchmark (e.g. Omukai+2000, Wakelam+2008, etc.)
- every test is described in the corresponding folder (look at the README file)
- not all the tests are working (check http://www.kromepackage.org/test/)

To run a test with KROME

```
$ ./krome -test=NAME_OF_THE_TEST
```

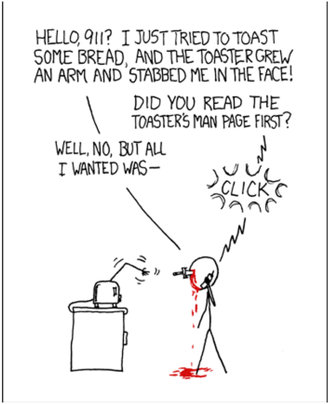
- Go into the "build", make and run!
- A gnuplot script is provided to plot the results, "load plot.gps".

Rules on chemical networks

and rates

1. functional form of the rates are always ugly (if not a Langevin!)
2. a complete network does not exist, by definition
3. never trust a pre-built network
4. if a rate is not wrong at least contains a typo
5. given a set of reaction rates at least one is bugged, including a set composed by one reaction rate!
6. networks for every season do not exist!

Additional info


HELLO, 911? I JUST TRIED TO TOAST SOME BREAD, AND THE TOASTER GREW AN ARM AND STABBED ME IN THE FACE!
DID YOU READ THE TOASTER'S MAN PAGE FIRST?
WELL, NO, BUT ALL I WANTED WAS—
CLICK

The information provided in this talk follow in part the Wiki page @https://bitbucket.org/tgrassi/krome/wiki/

www.kromepackage.org

Thank you for your attention!

KROME

better science through chemistry